

Compte rendu des travaux pratiques du cours de Java avancé N°2

Réalisé par : Intissar Lobad

Encadré par : Madame Asmaa Elkourchi

2024-2025

Table des matières

1. Architecture du Projet (MVC + DAO).....	3
1.1. Package Model.....	3
1.2. Package View.....	4
1.3. Package Controller.....	4
1.4. Package DAO.....	5
2. Fonctionnalités Analytiques.....	5
a) Ajouter un Congé (ajouterHoliday).....	5
b) Afficher les Congés (afficherHolidays).....	5
c) Modifier un Congé (modifierHoliday).....	6
d) Supprimer un Congé (supprimerHoliday).....	6
3. Fonctionnalités Supplémentaires.....	6
4. Conclusion.....	6

1. Architecture du Projet (MVC + DAO)

Le projet de gestion des congés est organisé en plusieurs packages respectant le design pattern **MVC (Model-View-Controller)**, complété par un module **DAO (Data Access Object)** pour gérer les interactions avec la base de données. Voici un aperçu des différents packages et leur contenu :

1.1. Package `Model`

Le package **Model** contient les classes métiers représentant les données et la logique métier du système.

- **Classe `Holiday` :**
 - Représente un congé avec les attributs suivants :
 - `id` : Identifiant unique du congé.
 - `startDate` et `endDate` : Dates de début et de fin du congé.
 - `holidayType` : Type de congé (ex : **CONGE_PAYEE**, **CONGE_NON_PAYEE**, **CONGE_MALADIE**).
 - `employeeId` et `employeeNom` : Identifiants et noms des employés liés au congé.
 - Les constructeurs permettent de créer des objets `Holiday` pour différentes opérations (affichage, ajout, modification).
- **Classe `HolidayModel` :**
 - Contient la logique métier et interagit avec la base de données via le DAO.
 - Principales méthodes :
 - `ajouterHoliday` : Ajoute un congé après validation des dates et du solde de l'employé.
 - `modifierHoliday` : Met à jour un congé existant.
 - `supprimerHoliday` : Supprime un congé en fonction de son identifiant.
 - `afficherHolidays` : Récupère tous les congés pour affichage.
 - `isValidDateRange` : Vérifie la validité de la plage de dates (début <= fin).

1.2. Package `view`

Le package **View** contient les interfaces graphiques (GUI) permettant l'interaction utilisateur.

- **Classe `HolidayView` :**
 - Fournit une interface utilisateur pour gérer les congés :
 - Champs pour les données de congé :
 - **Nom de l'employé** : Sélection via un `JComboBox`.
 - **Dates de début et fin** : Sélection via des `JSpinner`.
 - **Type de congé** : Choix via un `JComboBox`.
 - Tableau `JTable` :
 - Affiche les congés existants.
 - Permet de sélectionner un congé pour modification ou suppression via un double-clic.
 - Boutons d'action :
 - **Ajouter** : Ajoute un nouveau congé.
 - **Modifier** : Met à jour un congé sélectionné.
 - **Supprimer** : Supprime un congé sélectionné.
 - **Afficher** : Recharge les congés depuis la base.
 - Lorsqu'une ligne du tableau est double-cliquée, les champs sont automatiquement remplis avec les valeurs existantes pour permettre une modification.
- **Comportement additionnel :**

1.3. Package `Controller`

Le package **Controller** contient la logique qui connecte la vue au modèle.

- **Classe `HolidayController` :**
 - Interagit entre la vue (`HolidayView`) et le modèle (`HolidayModel`).
 - Actions principales :
 - **Ajout d'un congé** :
 - Récupère les données depuis la vue.
 - Valide les données et appelle `ajouterHoliday` du modèle.
 - **Affichage des congés** :
 - Appelle `afficherHolidays` du modèle et met à jour le tableau.
 - **Modification d'un congé** :
 - Récupère les nouvelles données via la vue.
 - Valide les dates et appelle `modifierHoliday`.
 - **Suppression d'un congé** :
 - Supprime un congé sélectionné via `supprimerHoliday`.

1.4. Package DAO

Le package **DAO** gère toutes les interactions avec la base de données.

- **Classe HolidayDAOImpl :**
 - Implémente les méthodes pour manipuler les données dans la table `holiday` :
 - `ajouter` : Ajoute un congé avec gestion du solde.
 - `afficher` : Récupère tous les congés avec les informations liées aux employés et types.
 - `modifier` : Met à jour un congé sélectionné.
 - `supprimer` : Supprime un congé en fonction de son ID.
 - Utilise `DBConnection` pour établir la connexion à la base.

2. Fonctionnalités Analytiques

a) Ajouter un Congé (`ajouterHoliday`)

- **Étapes détaillées :**
 1. **Récupération des données :**
 - Nom de l'employé, dates de début et fin, type de congé via la vue.
 2. **Validation métier :**
 - Vérifie que la date de début est antérieure ou égale à la date de fin.
 - Vérifie que le solde de congé est suffisant pour la durée demandée.
 3. **Insertion dans la base :**
 - Exécution d'une requête `INSERT INTO` avec calcul du solde restant.
 4. **Retour utilisateur :**
 - Affiche un message de succès ou d'échec dans la vue.

b) Afficher les Congés (`afficherHolidays`)

- **Étapes détaillées :**
 1. Connexion à la base et exécution d'une requête SQL pour récupérer les congés avec jointures sur les employés et types de congés.
 2. Mapping des résultats en objets `Holiday`.
 3. Mise à jour du tableau dans la vue via un `DefaultTableModel`.

c) Modifier un Congé (`modifierHoliday`)

- **Étapes détaillées :**
 1. Sélection dans le tableau :
 - Double-cliquer remplit les champs de la vue avec les données du congé sélectionné.
 2. Modification des données :
 - L'utilisateur peut changer les dates ou le type de congé.
 3. Validation :
 - Vérifie la validité des nouvelles dates.
 4. Mise à jour :
 - Exécution d'une requête `UPDATE` pour modifier l'enregistrement.

d) Supprimer un Congé (`supprimerHoliday`)

- **Étapes détaillées :**
 1. Sélection d'une ligne dans le tableau.
 2. Exécution d'une requête `DELETE` pour supprimer le congé de la base.

3. Fonctionnalités Supplémentaires

- **Chargement des noms d'employés :**
 - Les noms des employés sont récupérés via une requête SQL dans `chargerNomsEmployes`.
 - Les données sont affichées dans un `JComboBox` pour simplifier la sélection.
- **Messages utilisateurs :**
 - La vue affiche des messages clairs en cas d'erreur (dates invalides, solde insuffisant, etc.) ou de succès.

4. Conclusion

Le projet de gestion des congés suit une architecture propre et modulaire. Chaque couche est bien séparée et respecte les principes du MVC et DAO, facilitant la maintenance et l'ajout de nouvelles fonctionnalités.